

Contract-Based Quality of Service (QoS) Monitoring and Control of XML Web Services

Dr. Vladimir Tasic

Sch. of Computer Science
and Engineering
Univ. of New South Wales
Sydney, NSW, Australia

E-mail:

vladat (server: computer.org)

Prof. Patrick C.K. Hung

Faculty of Business and IT
Univ. of Ontario Institute
of Technology - UOIT
Oshawa, Ontario, Canada

Patrick.Hung (server: uoit.ca)

Slide
2 of 90

Tutorial Goals

- ⌘ Explain that QoS specification and management are **crucial** for achieving the vision of service-oriented computing (publish-find-bind model)
- ⌘ Inform that there have been **many** academic and industrial works in the area
- ⌘ Summarize and analyze the main past results
- ⌘ List and discuss open issues, such as **standardization** of specification and management of QoS for Web services (WSes)
- ⌘ Provide a **foundation** for future research and/or decision-making by the participants

Presentation Outline

- I. Introduction: Importance of QoS for WS
 - II. Approaches to specification of QoS for WS
 - III. Languages for specification of QoS for WS
 - IV. Approaches to management (monitoring and control) of QoS for WS
 - V. Research and industrial tools for management of QoS for WS
 - VI. Summary: Past results and open issues
- ⌘ Answers to questions and discussion

Slide
3 of 90

Module I:

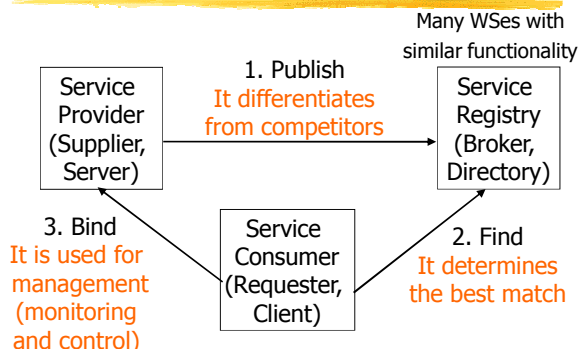
Definition of Terms and Importance of Web Service Quality of Service (QoS)

- ⌘ **Functionality/service** = "WHAT operations does the system execute?"
 - ☑ Example: Returns current price for a stock symbol
- ⌘ **Quality of service (QoS)** = "HOW WELL the system performs its operations?"
 - ☑ Examples: Average response time is 2 seconds, availability in the last 24 hours is 99%, ...
 - ☑ Synonyms: non-functional, extra-functional, 'ilities'
 - ☑ Price and security information sometimes included
 - ☑ QoS exists even when not specified or measured

Module
1 of VI

Slide
4 of 90

The Need for QoS Information in the Service-Oriented Architecture (SOA)



Module
1 of VI

Slide
5 of 90

Definition of Management - Monitoring

- ⌘ **Management = monitoring and control**
 - ☑ Run-time (and some deployment-time) activities
- ⌘ **Monitoring** determines state of the system:
 - ☑ Measurement or calculation of QoS metrics (measures of QoS): response time, availability, ...
 - ☑ Evaluation of conditions (requirements or guarantees): response time < 2 seconds, ...
 - ☑ Accounting of invoked operations, consumed resources, measured/calculated QoS metrics, evaluated conditions, taken control actions, billed prices/penalties, ...

Module
1 of VI

Slide
6 of 90

Module 1 of VI Slide 7 of 90

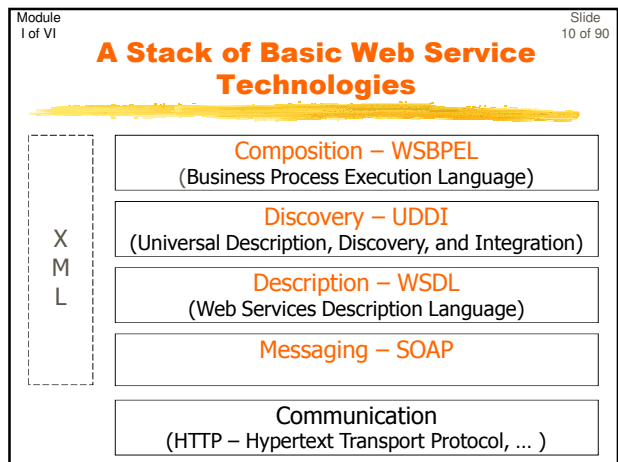
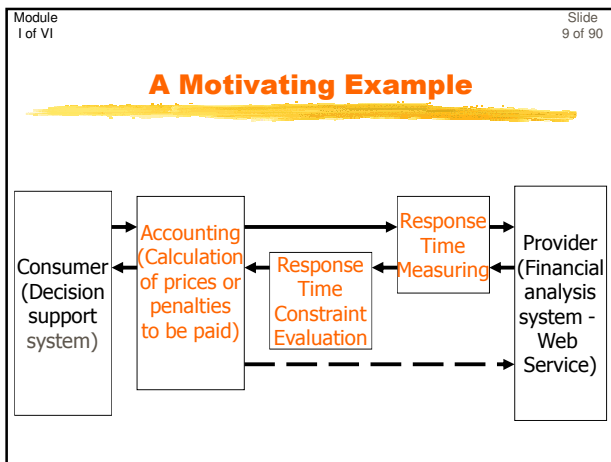
Definition of Management - Control

- ⌘ **Control** tries to ensure that the managed system is always in its desired state:
 - ☒ Starting/stopping the system or its components
 - ☒ (Re-)Configuration of the system: setting thread priorities, re-composition of Web services, ...
 - ☒ (Re-)Allocation of resources: assigning processing time to requests from different consumers, ...
 - ☒ Billing of prices or penalties: penalty for not meeting guaranteed response time is US\$1.00, ...
 - ☒ Modification of requirements or guarantees
 - ☒ Notification of human administrators

Module 1 of VI Slide 8 of 90

Benefits of QoS Management

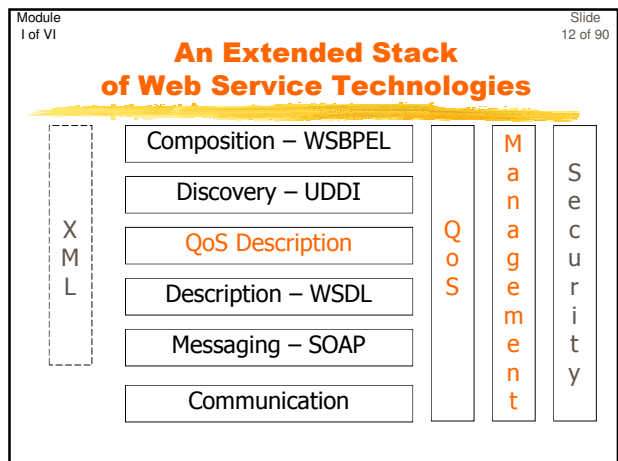
- ⌘ 5 functional areas of system/network management (FCAPS): Fault, Configuration, Accounting, **Performance**, and Security
- ⌘ **QoS (performance) management** helps to:
 - ☒ ensure correct operation,
 - ☒ attain or surpass guaranteed QoS,
 - ☒ discover and fix problems,
 - ☒ accommodate change,
 - ☒ balance price/performance ratios,
 - ☒ maximize profits, ...



Module 1 of VI Slide 11 of 90

The Place of QoS in the Stack of Web Service Technologies

- ⌘ Many additional technologies appeared
- ⌘ There are disagreements about the contents of the stack of Web service technologies
- ⌘ **Basic Web service technologies (SOAP, WSDL, UDDI, WS-BPEL) do not address QoS specification and management**
- ⌘ **Two approaches** to adding QoS specification and management to the stack
 1. Crosscutting different layers
 2. Adding a new layer



Module I of VI Slide 13 of 90

Examples of QoS Issues at Different Layers in the Stack

- ⌘ SOAP: "Exactly once" delivery, compression of SOAP messages, ... (not the focus of this tutorial)
- ⌘ WSDL: Describing QoS in addition to functionality (can be a separate layer in the stack), management of QoS of one WS, ...
- ⌘ UDDI: Discovery/selection of Web services with particular functionality and QoS, ...
- ⌘ WS-BPEL: Describing QoS dependencies between WSEs in a composition, management of QoS of the whole composition, ...

Module I of VI Slide 14 of 90

What Has to Be Developed for WS QoS Specification and Management?

1. Well-defined (ideally: standardized) **formats** for specification of WS QoS information
2. Many WS QoS-related **algorithms & protocols**
 - ☒ Very diverse: selection of WSEs using QoS info, negotiation of QoS of a WS, monitoring of QoS, exchange of run-time QoS info, control to achieve QoS guarantees, adaptation to changes in QoS, ...
 - ☒ We will discuss them in the context of specification formats and/or management tools
3. WS QoS **management infrastructures/tools**

Module II of VI Slide 15 of 90

Module II: Overview of Approaches to WS QoS Specification

- ⌘ QoS specification = description of what/where/when/how to monitor & control
- ⌘ QoS info = descriptions & monitored values
- ⌘ Classification of QoS specification **approaches**:
 1. **Implicit** – built into the implementation (not flexible)
 2. **Contracts** – formal agreements (for QoS, billing, ...)
 - ☒ Service Level Agreements (SLAs)
 - ☒ Classes of service – a special type of SLAs
 3. **Policies** – high-level operation & management goals and/or rules (for security, QoS, billing, ...)

Module II of VI Slide 16 of 90

Contract

- ⌘ **Contract** = binding and enforceable formal agreement between two or more parties
- ⌘ Defines **requirements & guarantees** of parties
 - ☒ Can be used in monitoring and control
- ⌘ Contracts enable not only **QoS description**, but also **QoS differentiation**
 - ☒ Different consumers can have different contracts
- ⌘ Apart from QoS info, a contract can contain **other information** (e.g., prices/penalties)
 - ☒ A WSDL file is a contract

Module II of VI Slide 17 of 90

Specification of QoS in Extended WSDL, UDDI, or WS-BPEL Files

- ⌘ **Strengths**:
 - ☒ The extensions can be relatively simple
 - ☒ QoS discovery related to Web service discovery
- ⌘ **Weaknesses**:
 - ☒ QoS specification language tied to WSDL (UDDI, WS-BPEL) in terms of tools, evolution, ...
 - ☒ Extension mechanisms are limited
 - ☒ **Run-time change of QoS information** requires updates of all affected copies of WSDL (UDDI, WS-BPEL) files, which is complicated

Module II of VI Slide 18 of 90

Service Level Agreement (SLA)

- ⌘ A special **type of contract** for QoS (and often price/penalty) requirements & guarantees
- ⌘ Many different **formats**, one of them is:
 - ☒ **Parties** (including supporting management parties)
 - ☒ **Service description**
 - ☒ Service operations – describe available operations
 - ☒ SLA parameters – define monitoring of QoS metrics
 - ☒ **Obligations**
 - ☒ **Service Level Objectives (SLOs)** - QoS guarantees
 - ☒ **Action guarantees** - specify what happens if SLOs are met or not met

Module II of VI Slide 19 of 90

A Simple Example of an SLA

Parties: consumer C and provider P

Service operations: P has one operation (OP1)
float getPrice(String stockName)

SLA parameters: (RT-OP1-C) Response time of operation OP1 measured at consumer C by consumer C

SLOs: (SLO1) For every OP1 invocation by C, RT-OP1-C will be less than or equal to 2 seconds

Action guarantees: (AG1) If SLO1 was met, C pays P price of US\$0.20 per invocation;
(AG2) If SLO1 was not met, P pays C penalty of US\$0.10 per invocation

Module II of VI Slide 20 of 90

Service Level Agreement (SLA) - Strength and Weaknesses

- ⌘ **Strengths:**
 - ☒ Formal contract specification of QoS and related management aspects
 - ☒ Widely used in computing and communications systems (and now also for WSes)
- ⌘ **Weaknesses:**
 - ☒ Negotiation of custom-made SLAs can require complex analysis of offers and generation of counter-offers (can be alleviated by using templates)
 - ☒ Management of many concurrent custom-made SLAs can be complex & with high run-time overhead
 - ☒ Cannot be used for QoS-enabled WS selection

Module II of VI Slide 21 of 90

Class of Service

- ⌘ A special type of SLA that is not custom-made, but predefined & reusable (anonymous)
 - ☒ 1 provider can offer many classes of service that refer to the same functionality, but differ in QoS
 - ☒ 1 class of service can be used by many consumers
 - ☒ Simple selection instead of complex negotiation
 - ☒ Classes of service already checked for consistency
 - ☒ Strengths: Usable for QoS-enabled WS selection, no complex negotiation, simpler management, lower run-time overhead, faster adaptation
 - ☒ Weakness: Discrete differentiation - limited choice

Module II of VI Slide 22 of 90

Policies - High-Level Goals and/or Rules

- ⌘ A classification of policy types [Kephart&Walsh2004]:
 - ☒ Action: Describe what should happen - "If-Then" rules ("If response time of operation A is greater than 2 sec, provider pays penalty of US\$0.10")
 - ☒ Goal: Describe desired state ("Response time of operation A is less than or equal to 2 sec")
 - ☒ Utility: Quantify "goodness" of a particular state ("Add to the goodness measure [2 sec - response time of operation A] * 10 units") - rarely used
- ⌘ SLAs vs. policies: SLOs can be viewed as goal policies, action guarantees as action policies

Module II of VI Slide 23 of 90

QoS Specification Topics Present in All Approaches

- ⌘ Where are QoS metrics defined?
 - ☒ There are no standard QoS metrics - use, names, and definitions vary! Example: 'response time' can have at least 2 different meanings! 4 approaches:
 - Nowhere (implicit meaning) - not precise
 - In the QoS language grammar - not flexible
 - In QoS specification files (e.g., SLAs) - not reusable
 - In external reusable ontologies (definition files)
 - ☒ Other ontologies can define measurement units
- ⌘ For practical use, QoS specification languages must be accompanied by appropriate tools!

Module II of VI Slide 24 of 90

Module II (Approaches to QoS Specification) - Summary & Discussion

- ⌘ Contract = binding and enforceable formal agreement between two or more parties
 - ☒ SLA is a special type of contract; many SLA formats
 - ☒ Class of service is a type of a light, predefined SLA
- ⌘ Which one to use depends on circumstances
 - ☒ For comprehensiveness: general contracts
 - ☒ For flexibility of QoS specification: custom-made SLAs
 - ☒ For low overhead: classes of service
- ⌘ Contracts vs. policies: similar information, different management architectures
 - ☒ Contract pluses: negotiation, use in WS selection, ...

Module III of VI Slide 25 of 90

Module III: Overview of XML Languages for WS QoS Specification

- ⌘ Classified based on the main concept:
 - **SLA**: WSLA, [SahaiEtAl2002] (WSML), SLAng
 - 1. **Class of service**: WSOL, WS-QoS, DAML-QoS
 - 2. **General contract**: OWL-S, WS-Agreement
 - 3. **Extension of UDDI/BPEL**: UDDIe, [McGregor2003]
 - 4. **Policy**: WS-Policy
 - 5. **Manageability capability**: WSDM
- ⌘ There are some other languages and formats, but these are probably best representatives

Module III of VI Slide 26 of 90

Why Not Reusing QoS Specification Languages from Other Areas?

- ⌘ Many existing QoS specification languages in multimedia (HQML, ...), distributed objects (QML, QDL, QIDL, ...), and other areas
- ⌘ Can not be directly re-used because of:
 - ☒ Incompatibility with WS standards (e.g., WSDL)
 - ☒ Heterogeneity of WS implementations and interactions styles used (asynchronous & synchronous; document-based & RPC; ...)
 - ☒ Characteristics of WS compositions: business-to-business, Internet scale, dynamism, automatism, ...

Module III of VI Slide 27 of 90

Web Service Level Agreement (WSLA) - Overview

from IBM Research: H. Ludwig, A. Keller, A. Dan, ...

- ⌘ QoS language & management infrastructure
- ⌘ Compatible with, but not restricted to WSEs
- ⌘ Custom-made SLAs that have 3 parts:
 1. **Parties** (signatory & supporting) and their management operations
 2. **Service definitions** – service objects (e.g., WSDL operations) and their monitored properties (QoS metrics, SLA parameters, schedules, triggers, ...)
 3. **Obligations** – SLOs and action guarantees

Module III of VI Slide 28 of 90

Web Service Level Agreement (WSLA) - Language Details

- ☒ **SLA parameter** - monitored property; contains 1 QoS metric & extra info for exchange of values
- ☒ **QoS metric** – defines where & how to measure or calculate; can be reused across SLA parameters
- ☒ An **SLO** contains: **evaluated Boolean expression** (limits values of SLA parameters), obliged party, validity periods, evaluation event or schedule
- ☒ An **action guarantee** contains: precondition expression, evaluation event or schedule, **action to be taken**, obliged party, execution modality
- ☒ Reusability: SLA templates, metric macros, ...

Module III of VI Slide 29 of 90

Web Service Level Agreement (WSLA) - Strengths and Weaknesses

- ⌘ **Strengths** (significantly overweight weaknesses):
 - ☒ **Detailed and precise** description of QoS monitoring and control
 - ☒ Several **tools** for SLA creation, deployment, and compliance monitoring – were distributed with IBM's Emerging Technologies Toolkit (ETTK)
 - ☒ The **most used** WS QoS specification language
- ⌘ **Weaknesses** (mainly due to custom-made SLAs):
 - ☒ QoS metrics defined within SLAs
 - ☒ Can not be used for QoS-enabled WS selection
 - ☒ High overhead of supporting custom-made SLAs

Module III of VI Slide 30 of 90

HP's Web Service SLA Language ([SahaiEtAl2002])

from HP Labs: A. Sahai, A. Durante, V. Machiraju, ...

- ☒ a.k.a. Web Service Management Language (WSML)
- ☒ for **Web Services Management Network (WSMN)**
- ⌘ **Custom-made SLAs** - Each contains: validity period, parties, and multiple SLOs
- ⌘ An **SLO** has: validity times & multiple clauses
- ⌘ A **clause** defines: monitored items (and places of their monitoring), evaluation times, samples, evaluation function, action to take
- ⌘ Similar to WSLA, but not as widely used

Module III of VI Slide 31 of 90

SLAng

from UC London: D.D. Lamanna, J. Skene, W. Emmerich

- ⌘ For **custom-made SLAs** for ASPs (WSEs) and underlying resources (horizontal/vertical SLAs)
- ⌘ Weaknesses: 1) Still "work in progress"; 2) **No management infrastructure**; 3) QoS metrics defined in language grammar – not flexible, hard to extend
- ⌘ Strengths: 1) Formally and very precisely defined semantics of the language and built-in QoS metrics; 2) Use meta-modeling to define language's syntax and semantics; 3) SLA conformance checker generated from the language definition

Module III of VI Slide 32 of 90

Web Service Offerings Language (WSOL) - Overview

from Carleton Univ.: V. Tasic, K. Patel, B. Pagurek, ...

- ☑ plus **Web Service Offerings Infrastructure (WSOI)**
- ☑ **Weakness**: parser available, but not a complete compiler generating monitoring code
- ⌘ **Classes of service** and their **relationships**
- ⌘ **Multiple categories of constraint**: functional, QoS (including periodic), access rights, context
 - ☑ Also: various prices and monetary penalties
- ⌘ QoS metrics defined in external ontologies
- ⌘ Its goal was relatively low run-time overhead

Module III of VI Slide 33 of 90

Web Service Offerings Language (WSOL) - Language Constructs

- ⌘ **Service Offering (SO)** - for a class of service
 - ☑ **constraints** (pre-/post-conditions, QoS guarantees, access rights, context constraints) contain expressions (Boolean, arithmetic, ...) to be evaluated
 - ☑ **statements** (prices, penalties, management responsibility, invocation retries, cache validity)
 - ☑ **reusability constructs** (extension, constraint groups, inclusion, constraint group template instantiation, ...)
- ⌘ **Service Offerings Dynamic Relationships (SODRs)** – if a set of constraints from current SO was not met, what is an appropriate replacement SO

Module III of VI Slide 34 of 90

WSDL and WSOL: An Example

Module III of VI Slide 35 of 90

Web Service Offerings Language (WSOL) - Syntax Example

```
<wsol:serviceOffering name="SO1" service="buyStock:buyStockService" accountingParty="WSOL-SUPPLIERWS">
...
<wsol:instantiate CGTName="CGT2" resService="..." resPortOrPortType="..." resOperation="..." resCGName="CG5">
  <wsol:paramValue name="maxResTime">
    <wsol:numberWithUnitConstant>
      <wsol:value>30</wsol:value>
      <wsol:unit type="QoSMeasOntology:millisecond"/>
    </wsol:numberWithUnitConstant>
  </wsol:paramValue>
</wsol:instantiate>
</wsol:serviceOffering>
```

Module III of VI Slide 36 of 90

Web Services Quality of Services (WS-QoS)

from Freie Univ. Berlin: M. Tian, A. Gramm, H. Ritter, ...

- ☑ plus **WS selection and monitoring infrastructure**
- ⌘ For **classes of service** on the WS level and network level (cross-layer, cross-domain approach)
- ⌘ Strength: QoS specified for **both** providers & consumers, so that they can be matched
- ⌘ Weakness: **Limited format** (no expressions, ...)
- ⌘ Informal definitions of several QoS metrics built into the language grammar, but new ones can be added through WS-QoS ontology

Module III of VI Slide 37 of 90

Web Ontology Language (OWL) - Services (OWL-S)

from the Semantic Web community

- ☒ Early version was called DAML-S
- ☒ Early version of OWL was called DAML, later DAML+OIL
- ⌘ Intended for **WS selection**, not QoS monitoring
- ⌘ **"Service profile"** contains comprehensive description of WSEs, including some QoS, functional constraints, prices, ...
- ⌘ Can be viewed as a general contract
- ⌘ Weaknesses: QoS specification is **very weak** – it is insufficient for monitoring activities

Module III of VI Slide 38 of 90

DAML-QoS – Overview

from Nanyang Tech.Univ.: C. Zhou, L.T. Chia, B.S. Lee

- ⌘ QoS **complement to DAML-S**; in DAML+OIL
- ⌘ **3 layers** with reusable info for QoS matching:
 1. **QoS profile** (provider/inquiry/template): contains scope, evaluation time; DAML+OIL cardinality is used to represent QoS property constraints; many QoS profiles per 1 DAML-S service profile
 2. **QoS property definition**: contains domain (input, output, ...), range (QoS metric)
 3. **QoS metric** (atomic/complex): monitoring info; some metrics built-in, others can be added

Module III of VI Slide 39 of 90

DAML-QoS - Discussion

- ⌘ **Strengths:**
 - ☒ Significant improvement over OWL-S (DAML-S)
 - ☒ Reuse of existing Semantic Web tools for reasoning
 - ☒ QoS specified for both providers and consumers
- ⌘ **Weaknesses:**
 - ☒ **No management infrastructure**
 - ☒ **Very hard to understand** – unusual design choices; it is not clear what monitoring information can or cannot be specified and how is this monitored
 - ☒ No support for control activities
 - ☒ Overhead of Semantic Web technologies

Module III of VI Slide 40 of 90

Web Services Policy Framework (WS-Policy)

from BEA/IBM/Microsoft/SAP – **industry support!**

- ⌘ **General, flexible and extensible, framework** for specification of (security) policies for WSEs
- ⌘ Many good features (e.g., policies can be in or out of WSDL files, some reusability constructs)
- ⌘ **QoS extension is possible, but missing! To add:**
 - ☒ Precise and detailed QoS specification
 - ☒ Contracts/SLAs/classes of service and their static and dynamic relationships
 - ☒ Standardized expression mechanism

Module III of VI Slide 41 of 90

WS-Agreement

from Global Grid Forum (GGF); industry support (IBM,...)

- ⌘ **General framework** for XML specification of agreements and agreement templates
 - ☒ plus a **simple agreement negotiation protocol and run-time agreement monitoring interface**
- ⌘ Intended for **multiple domains**, not only WSEs
- ⌘ WS-Agreement allows use of **any language** for the actual contained specifications (including QoS expressions, QoS metrics, ...)
- ⌘ This **flexibility** can produce **incompatibility**

Module III of VI Slide 42 of 90

WS-Agreement – Agreement Template Structure

- ⌘ **Name**
- ⌘ **Context:** Involved parties (initiator & provider); Expiration time; Template name, Related agreements
- ⌘ **Terms:** Term compositors ExactlyOne/OneOrMore/All
 - ☒ **Service description terms:** Service descriptions, Service references, Service properties
 - ☒ **Guarantee terms:** Service scope, Qualifying condition, **Service level objective (SLO)**, Business value list (Importance, Penalty, Reward, Preference, ...)
- ⌘ **Creation constraints:** item requirements and/or constraints (in some language)

Module III of VI Slide 43 of 90

Some Simple and Limited Languages - Extensions of UDDI, WSDL, or BPEL

- ⌘ **UDDIe (UDDI Extension)** – Adds to UDDI licensing information and a bag of arbitrary properties that can represent QoS metrics; QoS info also specified in extended WSDL files
from Cardiff Univ.: A. ShaikhAli, O.F. Rana, R.J. Al-Ali, ...
- ⌘ **Simple BPEL4WS extension** ([McGregor2003]) with several predefined QoS metrics to monitor and analyze business performance; Monitored QoS metrics and their target values
from Univ. of Western Sydney: C. McGregor

Module III of VI Slide 44 of 90

Web Services Distributed Management (WSDM)

from OASIS Web Services Distributed Management TC

- ☒ One of the used inputs: WS-Manageability
- ⌘ **Two sets of publications:**
 - ☒ **Management Using Web Services (MUWS)** - manageability interfaces of resources exposed as WSes
 - ☒ **Manageability capability** – set of operations, properties, events, metadata, and other info. Example: "metrics"
 - ☒ **Management Of Web Services (MOWS)** – WSes managed as resources and described with MUWS
 - ☒ Defines basic metrics, e.g., LastResponseTime
- ⌘ **Strengths:** Standard management interfaces
- ⌘ **Weaknesses:** Very limited QoS specification and management support; no specification of guarantees

Module III of VI Slide 45 of 90

Web Services Quality Model (WSQM)

from OASIS Web Services Quality Model TC

- ☒ started in October 2005, still "work in progress"
- ⌘ **Three phases and deliverables envisioned:**
 - 1) Conceptual **Web Services Quality Model (WSQM)**
 - 2) Its XML representation in the **Web Services Quality Description Language (WS-QDL)**
 - 3) **Test guidelines** for WSQM
- ⌘ **Strengths:** First standardization effort in this area, thorough approach to QoS issues
- ⌘ **Weaknesses:** Not yet completed, not a well-known effort, does not yet involve major industry players

Module III of VI Slide 46 of 90

WSQM - Main Concepts

- ⌘ **Components:** **Quality Factors** (metrics), **Quality Associates** (roles), and **Quality Actions** (activities)
- ⌘ **Three layers of Quality Factors** with several quality types (each of which can have sub-factors):
 - 1) **Business Value Quality** (Business Suitability, Business Effect, Business Recognition Level)
 - 2) **Service-Level Measurement Quality** (Performance, Stability)
 - 3) **System-Level Quality** (Interoperability, Business Processing, Manageability, Security)

Module III of VI Slide 47 of 90

Module III (Languages for QoS Specification) - Summary & Discussion

- ⌘ There are **many languages**; very different
 - ☒ Most are based on **contracts**, particularly SLAs
 - ☒ We presented several important ones, but emphasize:
- ⌘ **WS-Agreement & WS-Policy** as possible future general frameworks for WS QoS specification
 - ☒ Have industry support, but the "meat" is missing
 - ☒ We recommend aligning **WSQM** with (one of) them
- ⌘ **WSLA** as precise and detailed contract-based QoS specification language used in practice
 - ☒ Its solutions could be used (along with some ideas from other languages, e.g., WSOL) for the "meat"

Module III of VI Slide 48 of 90

Presentation Progress

- I. Introduction: Importance of QoS for WS
- II. Approaches to specification of QoS for WS
- III. Languages for specification of QoS for WS
- IV. => Approaches to management (monitoring and control) of QoS for WS
- V. Research and industrial tools for management of QoS for WS
- VI. Summary: Past results and open issues

- ⌘ Answers to questions and discussion

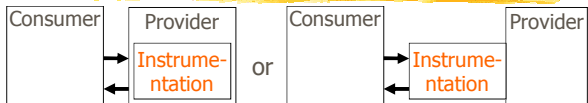
Module IV of VI Slide 49 of 90

Module IV: Overview of Approaches to WS QoS Management

- ⌘ Approaches to **QoS monitoring**
 - ☑ Instrumentation, intermediaries, probes, sniffers
- ⌘ Approaches to **QoS discovery and selection**
 - ☑ Using historical information vs. using contracts
 - ☑ Provider as only source of its QoS specifications, UDDI extensions, special QoS information registry
- ⌘ Some approaches to **QoS control**
 - ☑ Using different request queues for different QoS
 - ☑ Re-composition of Web services vs. re-negotiation of contracts

Module IV of VI Slide 50 of 90

Using Monitoring Instrumentation Only on the Provider Side



- ⌘ **Invasive vs. non-invasive instrumentation**
 - ☑ Invasive: within business logic code of a WS vs. within WS hosting tools (e.g., within SOAP engine)
- ⌘ All monitoring on the provider side
- ⌘ Strengths: **Realistic & consumer-specific measures; independent from network location of measurement**
- ⌘ Weaknesses: Provider must have capabilities and willingness; **consumers must trust the provider**

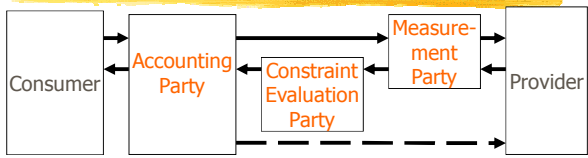
Module IV of VI Slide 51 of 90

Some Instrumentation Technologies

- ⌘ Industry standards:
 - ☑ Simple Network Management Protocol (SNMP)
 - ☑ Application Response Measurement (ARM)
 - ☑ Java Management Extensions (JMX)
 - ☑ Windows Management Instrumentation (WMI)
- ⌘ Other instrumentation approaches:
 - ☑ addition of composable SOAP message processing filters to WS hosting tools (e.g., a SOAP engine)
 - ☑ aspect-oriented code weaving (mostly research)
 - ☑ mobile agents (mostly research)
- ⌘ Most can be used not only on providers, but also on consumers and third parties

Module IV of VI Slide 52 of 90

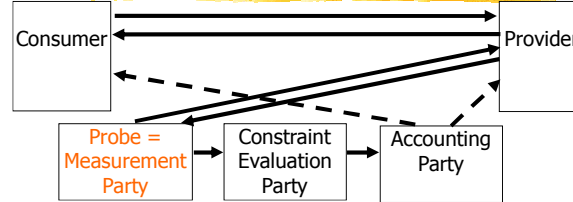
Using SOAP Message Intermediaries



- ⌘ Exchange of monitored values: a) in SOAP headers; b) using special push or pull operations
- ⌘ Strengths: **Realistic & consumer-specific measures**
- ⌘ Weaknesses: **High run-time overhead** (can be reduced with periodic/occasional monitoring); results depend on network location of measurement

Module IV of VI Slide 53 of 90

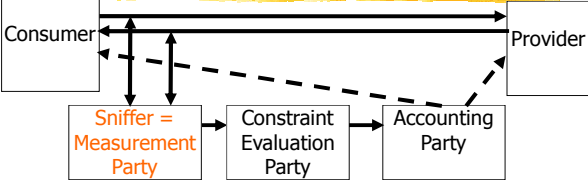
Using Probes (Probing)



- ⌘ Strengths: Run-time overhead can be lower
- ⌘ Weaknesses: **Results not consumer-specific, provider can treat probes in a special way; not possible to re-use SOAP headers to send monitored values; results depend on network location of probes**

Module IV of VI Slide 54 of 90

Using Sniffers (Sniffing)



- ⌘ Strengths: **Very low run-time overhead**; measures can be realistic & consumer-specific
- ⌘ Weaknesses: **Unknown SOAP message's Internet route; WS security technologies can be a problem; not possible to use SOAP headers to send monitored values; results depend on network location of sniffers**

Module IV of VI Slide 55 of 90

Using Historical QoS Information for WS Selection - Possible Approaches

- ⌘ From the **same consumer**
 - ☒ Problem: When consumer did not previously invoke this operation of the provider Web service
- ⌘ From **probes**
 - ☒ Problem: Easy for providers to give excellent QoS to probes, while bad QoS to real consumers
- ⌘ From **all consumers**
 - ☒ Problem: Consumers have different characteristics (e.g., could be located on different continents)
 - ☒ Problem: Other consumers' reports can be fake

Module IV of VI Slide 56 of 90

Using Historical QoS Information for WS Selection - Discussion

- ⌘ General problem: **Circumstances of different invocations are different!**
 - ☒ Example: When the number of provider's concurrent consumers grows, it is likely that QoS perceived by individual consumers will drop
- ⌘ General problem: **Absence of targets/goals to guide control activities** (including billing)
- ⌘ Conclusion: Historical QoS information can be **useful**, but it provides **no guarantees** (and can even be misleading) => **contracts are needed**

Module IV of VI Slide 57 of 90

Using Provider as the Only Source of Its QoS Specifications

1. Provider publishes **only its WSDL file** to the registry (e.g., UDDI registry)
2. Consumer searches registry **only based on functionality** and selects one Web service
3. Consumer and provider **negotiate** QoS (and prices/penalties) that consumer will receive

- ⌘ Strengths: **QoS can be dynamic - registry not updated when QoS changes**; QoS can be customer-specific; no need to extend UDDI
- ⌘ Weaknesses: **QoS not used for selection of WSes**

Module IV of VI Slide 58 of 90

Using UDDI Extensions Describing QoS - Explanation and Some Options

1. Provider publishes **its WSDL file and QoS specifications** to the extended UDDI registry
2. Consumer searches registry **based on both functionality and QoS** and selects one WS (and its QoS - if multiple classes of service)
3. Consumer binds to the **provider and its QoS**

- ⌘ Option: Extended UDDI can be a gateway to the real UDDI, which is not changed
- ⌘ Option: [Ran2003] suggests a QoS certifier - probe verifying QoS guarantees in the registry

Module IV of VI Slide 59 of 90

Using UDDI Extensions Describing QoS - Discussion

- ⌘ Strengths: **QoS used for selection of Web services; simple selection (not negotiation) of QoS**
- ⌘ Weaknesses:
 - ☒ **QoS changes require updating the registry** (most works update only stored QoS specifications)
 - ☒ **Consumers can have outdated QoS information** (unless all are informed about updates, which requires keeping info about all consumer-provider relationships and has high run-time overhead)
 - ☒ **Scalability can become an issue** if there are many QoS descriptions (e.g., classes of service) per WS

Module IV of VI Slide 60 of 90

Using Separate QoS Information Registry

```

    graph TD
      Provider[Provider] -- "1. Publish WSDL" --> UDDI[UDDI Registry]
      Provider -- "2. Publish QoS" --> QoS[QoS Registry]
      Consumer[Consumer] -- "3. Find WSes" --> UDDI
      Consumer -- "4. Find QoS" --> QoS
      Consumer -- "5. Select WS and its QoS" --> Provider
      Consumer -- "6. Bind" --> Provider
  
```

- ⌘ Several model variations (e.g., registry performs selection)
- ⌘ Strengths - same as for UDDI extensions, plus **no need to extend UDDI**
- ⌘ Weaknesses - same as for UDDI extensions, plus **higher run-time overhead and higher complexity**

Module IV of VI Slide 61 of 90

Several Control Approaches That Try to Meet QoS Guarantees

1. Manipulate which request is processed first
 - ☒ Provider has several different request queues, e.g., one for each class of service
 - ☒ Scheduler within the provider decides from which queue to process a request, depending on QoS guarantees, current load, queue lengths, ...
2. Manipulate thread priorities for different requests and/or OS scheduling discipline
3. General approach: Manipulate allocation of resources for various requests
4. Load balancing between replicas

Module IV of VI Slide 62 of 90

Re-composition of Web Services vs. Re-negotiation of Contracts

- ⌘ Run-time adaptation of WS compositions
 - a) Re-composition of Web services – more powerful
 - ☒ Special case: Switching between WSeS (only 1 change)
 - b) Re-negotiation of contracts – faster, simpler, lighter
 - ☒ Special case: Switching between classes of service

Legend: C – consumer; P – provider; CS – class of service

Module IV of VI Slide 63 of 90

Mechanisms for Run-Time Manipulation of Classes of Service

1. Switching between classes of service (CSes)
 - ☒ Provider-initiated or consumer-initiated
2. Deactivation and reactivation of CSes
 - ⌘ With accommodation of affected consumers
3. Deletion of deactivated CSes
4. Creation of new CSes (same WSDL files)
 - ☒ Provider decides when it is possible and allowed
5. Allowing/disallowing consumers to use a CS
 - ⌘ Not always a replacement for the alternatives (incl. re-negotiation of SLAs), but can be their simple, fast, and lightweight complement

Module IV of VI Slide 64 of 90

Some QoS Issues at the SOAP Level

- ⌘ Reliable delivery of SOAP messages (e.g., “exactly once” delivery)
 - ☒ WS-Reliability – OASIS Web Services Reliable Messaging (WSRM) TC (originally: Fujitsu, Hitachi, NEC, Oracle, Sonic Software, Sun Microsystems)
 - ☒ WS-ReliableMessaging – OASIS Web Services Reliable Exchange (WS-RX) TC (originally: BEA Systems, IBM, Microsoft, TIBCO Software)
- ⌘ Compression of SOAP messages to reduce network transmission
 - ☒ Several approaches to XML compression

Module IV of VI Slide 65 of 90

Module IV (Approaches to QoS Management) - Summary & Discussion

- ⌘ Several different approaches to QoS monitoring, selection, and control
 - ☒ None is best for all circumstances – knowing their advantages/disadvantages will help you choose
- ⌘ In B2B scenarios, QoS monitoring with SOAP message intermediaries seems most flexible
- ⌘ For QoS-aware WS selection, contracts have significant advantages over historical QoS data
- ⌘ QoS control by (re-)allocating resources to meet QoS guarantees is necessary, but hard

Module V of VI Slide 66 of 90

Module V: Research and Industrial Tools for WS QoS Management

- ⌘ Published as research work in refereed papers
 - ☒ QoS-enabled selection of WSeS: UDDIe, UX
 - ☒ Contract-based QoS management: WSLA, WSMF ([SahaiEtAl2002]), WSOI (WSOL), WS-QoS, Cremona (WS-Agreement)
 - ☒ Other: Smartware, ... and many, many others
- ⌘ Commercial products (non-refereed literature)
 - ☒ WS QoS management products from specialized companies
 - ☒ Large systems management suites
 - ☒ Related management products

Module V of VI Slide 67 of 90

UDDI Extensions: UDDIe and UX

- ⌘ **UDDIe (UDDI Extension)**
from Cardiff Univ.; uses UDDIe data format (slide 43)
 - ☒ New operations for corresponding UDDIe search
 - ☒ Lease manager and lease-related operations
 - ☒ Used in **G-QoS architecture** for Grid computing
- ⌘ **UX (UDDI eXtension)**
from Nanyang Tech.Univ.: C. Zhou, L.T. Chia, B.S. Lee
 - ☒ Acts as a gateway to the standard UDDI
 - ☒ Contains historical QoS information reported by consumers and probes ("test hosts")
 - ☒ QoS metrics defined implicitly

Module V of VI Slide 68 of 90

Some Other Published Works on QoS-Enabled WS Discovery and Selection

- ⌘ **Quality of Compliance (WS-QoC)** – monitors SLAs and calculates normalized weighted difference between QoS guarantees and monitored values
from Cardiff Univ.: A. ShaikhAli, O.F. Rana, D. Walker
- ⌘ **[Day & Deters 2004]** – consumers collect QoS info and report it to QoS info registry ("QoS forum"); rule-based and naive Bayes reasoners for WS selection
from Univ. of Saskatchewan: J. Day, R. Deters
- ⌘ **[DeoraEtAl2003]** – compare user ratings and expectations
from Cardiff Univ.: V. Deora, J. Shao, W.A. Gray, N.J. Fiddian
- ⌘ **Solution Manager Service** – QoS info registry for [McGregor2003] extension of BPEL4WS (slide 43)

Module V of VI Slide 69 of 90

Web Service Level Agreement (WSLA) Framework

from IBM Research; uses WSLA language (slides 27-29)

- ⌘ **Modules = services:** 1) Establishment; 2) Deployment; 3) Measurement; 4) Condition Evaluation; 5) Management; 6) Business Entity
 - ☒ Prototype: **SLA Compliance Monitor** – module 1 is simple, 2 is implemented, 3 & 4 are general purpose, 5 & 6 missing
 - ☒ Service Deployment Information (SDI) is a subset of WSLA
 - ☒ Special management port types (e.g., for value exchange)
- ⌘ **Strengths:** The most widely used WS QoS research infrastructure; comprehensive approach to QoS management; support for management third parties
- ⌘ **Weaknesses:** Run-time overhead

Module V of VI Slide 70 of 90

Web Services Management Network (WSMN)

from HP Labs; uses [SahaiEtAl2002] language (slide 30)

- ⌘ **Main element: WSMN intermediary**
 - ☒ A proxy between the Web service and the outside
 - ☒ **Monitors SLAs** and exchanges management information using lifecycle, measurement, and assurance (e.g., negotiation) protocols
 - ☒ Run-time QoS info exchanged using protocols
 - ☒ General purpose modules within a SOAP engine
 - ☒ Prototype: **Business Management Platform (BMP) Agent** implemented over Apache SOAP (predecessor of Axis)
- ⌘ **Strengths:** Enables SLA-based QoS management
- ⌘ **Weaknesses:** Complexity & run-time overhead

Module V of VI Slide 71 of 90

Web Service Offerings Infrastructure (WSOI)

from Carleton Univ.; uses WSOL language (slides 32-35)

- ⌘ Extends open-source **Apache Axis** SOAP engine, run over Apache Tomcat app server
- ⌘ **Monitoring** of WSOL service offerings
- ⌘ **Dynamic manipulation** of service offerings
- ⌘ **Strengths:** Unique management support for WS classes of service, relatively low run-time overhead
- ⌘ **Weaknesses:** Limited control activities to meet QoS guarantees, not all modules implemented in the prototype, incomplete documentation

Module V of VI Slide 72 of 90

Web Service Offerings Infrastructure (WSOI) – Main Modules

- ⌘ **Modules for monitoring:**
 - ☒ Standard Axis modules (including handlers and chains)
 - ☒ **WSOI-specific handlers and chains (QoS monitoring)**
 - ☒ **WSODEngine** modules (for ordering of WSOI handlers)
 - ☒ **Timer** (for periodic activities)
- ⌘ **Modules for manipulation:**
 - ☒ **SOMgmtDecisions** (code of management algorithms)
- ⌘ **Modules for both monitoring & manipulation:**
 - ☒ Modules for **Service Offering Management (SOM)** port types (used in management protocols)
 - ☒ **Data structures** (service offering descriptions, consumer info, accounted management info within sessions, ...)

Module V of VI Slide 73 of 90

WSOI Inside a Provider Web Service

- ☑ Can also be used in consumers and SOAP intermediaries
- ⌘ Exchange of management (incl. QoS) info: in SOAP headers or using special built-in push or pull operations (one of SOM port types)

```

    graph LR
      subgraph Provider
        Tomcat[Tomcat] -- SOAP --> WSOI[WSOI (extends Axis)]
        WSOI -- op. call --> Code[Code of operations]
      end
      Consumer[Consumer] -- SOAP over HTTP --> Tomcat
  
```

Module V of VI Slide 74 of 90

Example of Using WSOI: WSOI-specific Handlers and Chains

```

    graph LR
      req[req.] --> SH[SH]
      SH --> SOI[SOI]
      subgraph WSOIChain [WSOIChain]
        AI[AI]
        RTB[RTB]
        RTS[RTS]
        QCE[QCE]
        AO[AO]
        SOO[SOO]
      end
      SOI --> AI
      AI --> RTB
      RTB --> Pr[Pr]
      Pr --> RTS
      RTS --> QCE
      QCE --> AO
      AO --> SOO
      SOO --> SH
      Pr --> opcall[op. call]
  
```

Module V of VI Slide 75 of 90

Web Services Quality of Service (WS-QoS) Infrastructure

from Freie Uni. Berlin; uses WS-QoS language (slide 36)

- ⌘ QoS-enabled WS discovery and selection
 - ☑ "QoS broker" – a variant of QoS info registry that (pre-)fetches QoS info and selects WSEs and QoS
- ⌘ QoS monitoring – (mostly) by provider; "QoS channel" in SOAP headers = consumer sends metrics to measure, provider sends measured values
 - ☑ "QoS proxy" – between Web service and network
- ⌘ Strengths: Deals with both QoS-aware WS discovery/selection and QoS monitoring
- ⌘ Weaknesses: Assumes QoS-aware network layer

Module V of VI Slide 76 of 90

CREation and MONitoring of Agreements (Cremona)

from IBM Research; uses WS-Agreement (slides 41-42)

- ⌘ Architecture for WS-Agreement middleware
 - ☑ Agreement initiator & agreement provider roles
 - ☑ Agreement Management layers: a) ... Protocol Role ... (APRM), b) ... Service Role ... (ASRM), c) Strategic ... (SAM)
- ⌘ Java library that: 1) implements WS-Agreement interfaces; 2) provides management functionality for agreement templates and instances; 3) defines abstractions to be implemented in domain-specific environments
- ⌘ Strengths: Relates agreements with underlying resources; reusable for various domains
- ⌘ Weaknesses: Needs additions to be used for WSEs

Module V of VI Slide 77 of 90

Smartware

from Infosys: A. Sharma, H. Adarkar, S. Sengupta

- ⌘ QoS control: Differentiated scheduling of requests based on context priorities
 - ☑ Context = info about provider application, user, and client device; sent by consumer in request SOAP header
- ⌘ Based on Apache Axis SOAP engine, adds:
 - ☑ Interceptor – reads context info and determines priority
 - ☑ Scheduler – puts request into a queue for its priority; based on scheduling policy fetches a request from a queue
 - ☑ Dispatcher – forwards request to the provider
- ⌘ Strengths: Rare work that performs QoS control
- ⌘ Weaknesses: Scheduling uses limited information

Module V of VI Slide 78 of 90

Computational Quality Attributes (CQA) Processing Service

by C.K. Fung, P.C.K. Hung, R.C. Linger, G.H. Walton

- ⌘ Flow-Service-Quality (FSQ) engineering
- ⌘ Key component services: QoS Manager, Establishment S., Policy Manager, Resource Manager, Prediction S., Operation S., Maintenance S., Monitoring S., Adaptation S., Diagnostic S.
- ⌘ BPEL extended with WSLA expressed in CQA
- ⌘ Strengths: New approach to specification and evaluation of dynamic service- and flow-level constraints & QoS using Bayesian statistical models
- ⌘ Weaknesses: Monitoring is by probing only

Module V of VI Slide 79 of 90

Some Other Published Works on QoS Management (slide 1 of 2)

- ⌘ **wsBus** – WS middleware; intermediary between providers and consumers; provides message prioritization, reliability, fault tolerance, load balancing, and security from Univ. of New South Wales: A. Erradi, P. Maheshwari, ...
- ⌘ **[Yu & Lin 2005]** – “QoS broker” with workflow & QoS information; implements dynamic switching of WSEs from Univ. of California – Irvine: T. Yu, K.J. Li
- ⌘ **[Baresi & Guinea 2005]** – runtime monitoring of WS-BPEL processes; proxy-based and rule-base approach from Politecnico di Milano: L. Baresi and S. Guinea
- ⌘ **PEM** – process execution management with runtime optimization based on business value metrics from IBM Research: M.J. Buco, R.N. Chang, L.Z. Luan, E. So, ...

Module V of VI Slide 80 of 90

Some Other Published Works on QoS Management (slide 2 of 2)

- ⌘ **Multichannel Adaptive Information Systems (MAIS)** – not WSEs, but the adaptation can be applied to WSEs from Univ. Roma & Pol. Milano: C. Marchetti, B. Pernici, P. Plebani
- ⌘ **Web Services Management Layer (WSML)** – uses aspect-oriented programming to code WS monitoring from Vrije Univ. Brussel: B. Verheecke, M.A. Cibrán, V. Jonckers
- ⌘ **Verity** – mechanisms and architecture to check and express provider compliance with QoS guarantees from Monash Univ.: S. Kalepu, S. Krishnaswamy, S.W. Loke
- ⌘ **???** – workshop participants summarize here other published works on QoS management from academia, industry, and/or government labs

Module V of VI Slide 81 of 90

Some Observations about Industrial Products for WS QoS Management

- ⌘ They **address many practical problems**
 - ☒ Academic researchers should be aware of these works and their accomplishments
 - ☒ Some works contain advanced solutions that show how SLAs and/or policies can be used in practice
- ⌘ Many products have significant **limitations**:
 - ☒ Crucial role of human administrators (i.e., not completely automated)
 - ☒ Limited/predefined choice of used QoS metrics
 - ☒ Lack of flexible formal machine-understandable QoS specification (instead, forms are used)

Module V of VI Slide 82 of 90

Some Products for WS QoS Management from Specialized Companies

- ☒ Often products (1 or more) addressing several management areas, including performance (QoS)
- ⌘ **Actional SOA Management** (including SOAPStation Web Services Broker) – policies
- ⌘ **AmberPoint** (including Service Level Manager) – custom-made SLAs
- ⌘ **Blue Titan** (including Network Director) – policies
- ⌘ **Infravio X-Registry** (including Management Console) – custom-made SLAs
- ⌘ **WestGlobal mScope** (including Performance Management Module - PMM) – custom-made SLAs

Module V of VI Slide 83 of 90

Large System Management Suites

- ⌘ **Contain many different management products**
 - ☒ some related to WSEs (or “business services”)
 - ☒ some related to performance (QoS) management of applications, computing systems, networks
- **HP: OpenView** (includes **SOA Manager**)
- 1. **IBM: Tivoli** (includes Business Systems Manager)
- 2. **Computer Associates (CA): Unicenter** (includes **Web Services Distributed Management – WSDM**)
- 3. **BMC Software: Patrol** (includes MAINVIEW)
- 4. **Microsoft** (includes Application Center)

Module V of VI Slide 84 of 90

Some Related Management Products

- ⌘ **Web service infrastructures, such as:**
 - ☒ **IONA Artix** (very limited support for QoS)
 - ☒ **webMethods Enterprise Services Platform**
- ⌘ **Application performance tools, such as:**
 - ☒ Many tools use Java Management Extensions (JMX)
 - ☒ **Mercury**: Service Level Management, ...
 - ☒ **OPNET (Altaworks)**: Commander, Panorama, ...
 - ☒ **Quest Software**: PerformaSure, Foglight, ...
- ⌘ **Web service security management tools**
- ⌘ **???** - workshop participants add here other products related to QoS management of Web services

Module V of VI Slide 85 of 90

Module V (Tools for QoS Management) - Summary & Discussion

- ⌘ There are **many tools**; very different in power
 - ☑ Most are based on **SLAs**, some are based on policies
- ⌘ **No current tool or a set of tools addresses all QoS monitoring, selection, and control needs!**
- ⌘ **Industrial products** address many basic issues, but have limitations (e.g., in QoS control)
 - ☑ Powerful, but expensive: **system management suites**
- ⌘ **Research tools** tackle advanced problems, but commercial use requires additional features
 - ☑ **WSLA Framework** is the most widely used of them

Module VI of VI Slide 86 of 90

Module VI: Summary of QoS Specification and Management for Web Services

- ⌘ QoS & manageability (also price, security, trust, ...) will be **differentiators** in the WS market
 - ☑ Determine WS to be chosen among similar ones
- ⌘ **QoS specification is the basis for management**
- ⌘ **Management (monitoring and control) is necessary** to meet QoS guarantees, discover and fix problems, accommodate change, ...
- ⌘ The vision of **service-oriented computing can not be achieved without (QoS) management**
 - ☑ But, the basic Web service technologies do not address QoS specification and management

Module VI of VI Slide 87 of 90

The Main Achieved Results

- ⌘ Several ...
 - ☑ ... **languages** for precise and detailed specification of contracts, SLAs, classes of service for Web services
 - ☑ ... **general frameworks** that can be extended for QoS specification (WS-Agreement, WS-Policy, OWL-S)
 - ☑ ... **industrial standards** (WSQM, WSDM, WS-Reliability, ...)
 - ☑ ... prototyped **solutions for QoS-enabled Web service discovery/selection**
 - ☑ ... research **infrastructures that enable WS QoS management** (predominantly monitoring)
 - ☑ ... **industrial products** used in practice

Module VI of VI Slide 88 of 90

Some Open Research Topics (slide 1 of 2)

- ⌘ **Standards for Web Service QoS specification**
 - ☑ QoS specification is no longer a research issue
 - ☑ **Contracts vs. policies**
 - ☑ Which existing work to extend into a standard? WSDM? WS-Agreement? ...
 - ☑ Equal status of requirements and guarantees
- ⌘ **Automatic negotiation of QoS contracts**
- ⌘ **Standards for WS QoS management interfaces**
 - ☑ Which existing work to extend into a standard? WSDM? WS-Agreement? ...
- ⌘ **Trust/reputation management related to QoS**

Module VI of VI Slide 89 of 90

Some Open Research Topics (slide 2 of 2)

- ⌘ **Control of Web Services to meet guarantees**
 - ☑ The **main area** for near-future research (on the contrary, monitoring is mostly well-researched)
 - ☑ Resource capacity planning and management
 - ☑ Building complex control plans
- ⌘ Solutions for **adaptation to various changes**
- ⌘ **Integrated management** of business operations, Web services, and underlying computing/communication infrastructure
 - ☑ Standard models of operation & issues at different levels and mappings between them are needed

Module VI of VI Slide 90 of 90

Resources

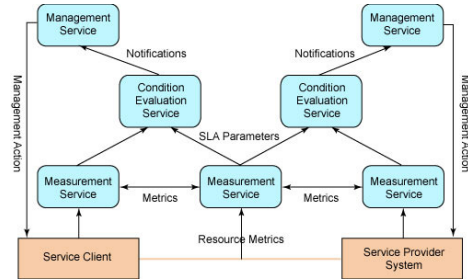
- ⌘ **Publications** are scattered between many different conferences, journals, and books
 - ☑ **Specialized workshops** (e.g., WQW 2003/2004, WISQ 2005, CoLa 2004/2005, MWS 2005) contain many relevant papers on this particular topic
 - ☑ **Survey papers** (e.g., [Tang et al. 2005]) give overviews
- ⌘ Our **annotated bibliography** available at: <http://www.cse.unsw.edu.au/~vtosic/QoS4WS-CBM4WS.doc>
- ☑ All works mentioned in this tutorial are listed
- ⌘ **Ask the tutorial presenters** (e-mail on Slide 1)
- ⌘ **Hire one of the tutorial presenters!** (Knowledge, experience, hard work, dedication, passion, ...)

Comparison of Some Web Service Contractual Description Languages

Language	Contract Category	Contract Type	Identity	Functionality				Quality		Infrastructure	
				Syntactic	Behavioral	Synchronization	Compositional	QoS	Pricing	Communication	Security
WSDL			+	+							+
WSBPPEL						+	+				
WS-CDL						+	+				
WS-Policy					+						+
WSLA								+	+		+
WSOL					+			+	+		+
OWL-S			+	+	+	+					

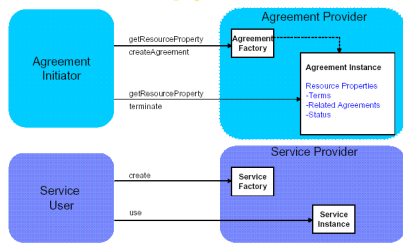
Reference/source: V. Tosic, B. Pagurek: On Comprehensive Contractual Descriptions of Web Services. In Proc. of the 2005 IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE-05), Hong Kong, China, March 29 – April 1, 2005, IEEE, pp. 444 - 449

Web Service Level Agreement (WSLA) Framework - Run-Time Use



Reference/source: Dan, A., Davis, D., Kearney, R., Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M., Youssef, A.: Web Services on Demand: WSLA-Driven Automated Management. IBM Systems Journal, Vol. 43, No. 1. IBM (2004) 136-158.

CREation and MONitoring of Agreements (Cremona) - Agreement Roles



Agreement roles independent of service roles

Reference/source: H. Ludwig, A. Dan, B. Kearney: Cremona: An Architecture and Library for Creation and Monitoring of WS-Agreements. In: M. Aiello, M. Aoyama, F. Curbera, M. Papazoglou: Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC 2004), New York, USA, December 2004, ACM Press, pp. 65 - 74